Online Nonnegative Canonical Polyadic Decomposition: Algorithms and Application

Isaac Wilfried Sanou
Université de Toulon, Aix Marseille Université, CNRS/INSU, IRD, MIO UM 110, CNRS, LIS, UMR 7020,
F-83957, CS 83041 La Garde, France
isaac-wilfried-sanou@etud.univ-tln.fr

Roland Redon
Université de Toulon, Aix Marseille Université, CNRS/INSU, IRD, MIO UM 110,
CS 83041 La Garde, France
roland.redon@univ-tln.fr

Xavier Luciani
Université de Toulon, Aix Marseille Université, CNRS, LIS, UMR 7020
F-83957 La Garde, France
xavier.luciani@univ-tln.fr

Stéphane Mounier
Université de Toulon, Aix Marseille Université, CNRS/INSU, IRD, MIO UM 110,
CS 83041 La Garde, France
stephane.mounier@univ-tln.fr

***Abstract*—The Nonnegative Canonical Polyadic Decomposition (NN-CPD) is now widely used in signal processing to decompose multi-way arrays thanks to nonnegative factor matrices. In many applications, a three way array is built from collections of 2D-signals and new signals are regularly recorded. In this case one may want to update the factor matrices after each new measurement without computing the NN-CPD of the whole array. We then speak of Online NN-CPD. In this context the main difficulty is that the number of relevant factors is unknown and can vary with time. In this paper we propose two algorithms to compute the Online NN-CPD based on sparse dictionary learning. We also introduce an application example of Online NN-CPD in environmental sciences and evaluate the performances of the proposed approach in this context on real data.**

***Index Terms*—Third order tensor decomposition, PARAFAC, online tensor decomposition, sparsity, dictionary learning, fluorescence signal processing**

I. Introduction

Multidimensional analysis and tensor decomposition such as the Canonical Polyadic Decomposition (CPD) also known as PARAFAC allows to decompose multi-way data into factors that can be interpreted by the user [1]. Hence tensor decompositions have been successfully applied in many fields including psychometric [2], neuroscience [3], chemometric [4], biomedical image processing [5], etc. In the last decades many iterative algorithms have been proposed to solve this decomposition problem. We can notably cite alternating approaches such as Alternating Least Squares (ALS) [2] or more traditional gradient or second order descent methods [6]. In several applications such as fluorescence spectroscopy the decomposition factors are known to be nonnegative [4]. This information can then be used to help algorithms convergence. Indeed nonnegative constraints can be easily implemented using a penalty term or a projected approach. This is for instance the case of the AO-ADMM algorithm [7].

In signal or data processing, the nonnegative CPD is usually used as a mathematical model to fit a data tensor $\boldsymbol{\mathcal{T}}$ of size (I, J, K):

$$\forall i, j, k, \quad \mathcal{T}_{i,j,k} \simeq \widehat{\mathcal{T}}_{i,j,k}(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \sum_{r=1}^{R} \mathbf{A}_{ir}\mathbf{B}_{jr}\mathbf{C}_{kr} \qquad (1)$$

where matrices $\mathbf{A} \in \mathbb{R}_+^{I\times R}$, $\mathbf{B} \in \mathbb{R}_+^{J\times R}$ and $\mathbf{C} \in \mathbb{R}_+^{K\times R}$ are the so-called factor or loading matrices. Columns of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ define the CPD factors and R is the CPD rank. When the CPD factors have a physical interpretation, it is important to distinguish between two particular values of R:

- The value of R for which the CPD is exact. This value defines the rank of $\boldsymbol{\mathcal{T}}$.
- The maximal value of R for which all the factors have a physical interpretation. We call this value the physical rank of $\boldsymbol{\mathcal{T}}$. It is usually much smaller than the tensor rank.

In practice, we only use the physical rank, that is why in the following, the term "rank" will always refer to the physical rank. Most CPD algorithms assume that the rank is known and use it as an input. However, in many practical applications we only know a magnitude order. Using a wrong value of the rank does not only affect the number of estimated factors but also their shape. In order to deal with this issue two main approaches have been proposed: rank estimation and overfactoring. Rank estimation consists in estimating the appropriate CPD rank before the decomposition by an $adhoc$ method such as the CORe CONsistency DIAgnostic [8] or AutoTen [9]. However, this approach does not always allow to clearly decide between several values. The overfactoring approach consists in overestimating the CPD rank during the decomposition in such a way that the contribution of the estimated extra-factors to the model will be negligible. Overfactoring can thus be seen as a posterior rank estimation method. For this purpose, several CPD algorithms have been modified to include some sparsity constraints upon the factor matrices using the L_1 norm [10]–[12], the mixed norm [13] or sparse known dictionaries [14]. Eventually, in [15], authors showed that semi-algebraic algorithms based on SVD and joint diagonalization were robust to rank overestimation in practice. At this stage we have implicitly assumed that the CPD is used to decompose the whole data tensor. In this work we consider a slightly different situation. We firstly assume that the data tensor is a 3-way array that gathers on its last mode a collection of 2D signals and that new collections of signals

are recorded regularly. Each collection of a new set of 2D signals is called a sub-tensor. Note that we can consider an overlapping between consecutive sub-tensors as in Fig. 1 or no overlapping. In other words, the last dimension (mode $\mathbf{C}$) of the data tensors grows with time. We also assume that the rank of the tensor and the sub-tensors are unknown and vary with time so that the CPDs of two successive sub-tensors do not necessary share the same factors in $\mathbf{A}$ and $\mathbf{B}$. On one hand the data tensor can become very large so that its CPD could be very time consuming. On the other hand one may not want to compute the CPD of the sub-tensors independently since we can reasonably assume that some factors are common to several sub-tensors. Our goal is then to update at each new time interval the CPD factors previously estimated, without performing the CPD of the whole tensor and taking into account possible disappearance and/or appearance of some factors. We call this problem Online CPD. Similar problems have been investigated in the literature. In [16] and [17], authors compute the CPD of successive sub-tensors in an adaptive way but they assume that the rank is known and does not vary with time while in [18] the sub-tensors are treated independently using AutoTen for rank estimation. Eventually, in [19] authors introduced an Online Tucker decomposition.

In the first part of this paper (section II), we propose two algorithms based on overfactoring and using sparse dictionary learning to compute the NonNegative Online CPD (NN-CPD) of a third order tensor[1]. Here, dictionary learning allows us to use the factors estimated at time t_{n-1} to compute the CPD of the current sub-tensor $\mathcal{T}_n$ while a sparsity constraint on the atoms allows us to overestimate the rank. Our two algorithms differ in the way the dictionaries are updated between two time intervals. In the second part (section III) we show how the Online CPD can be used for fluorescence signal processing in the context of environmental sciences. Our algorithms are then evaluated thanks to real fluorescence data.

In the rest of this article, the following notations will be used: scalars are denoted in italic letters, matrices in bold capital letters and tensors in calligraphic bold letters. $||.||_F$ is the Frobenius norm of a matrix or a tensor. $\mathbf{A}, \mathbf{B}, \mathbf{C} \geq 0$ means that all the elements of these matrices are nonnegative.

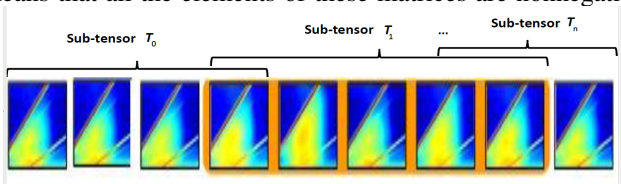

Figure 1. Example of online acquisition of 2D signals with overlapping.

II. ONLINE NN-CPD ALGORITHMS

We introduce two algorithms for the Online NN-CPD problem described in introduction. Both algorithms have two phases: the *initialization* (section II-A) phase and then the *online* phase (section II-B). In the *initialization* phase, we estimate nonnegative factor matrices without any prior information. This phase is identical for both algorithms. Estimated factor matrices $\mathbf{A}$ and $\mathbf{B}$ will then be used in the *online* phase. Furthermore, we assume that we know an upper bound of the physical rank and for each phase we use a fixed CPD rank, R, greater or equal to this bound. We denote $\widetilde{R}_n$ the physical rank of the sub-tensor $\mathcal{T}_n$ measured a time t_n. Both algorithms consist in solving optimization problems iteratively. The optimization scheme is the same for both algorithms and it is given in section II-C.

A. Initialization *phase. Tensor decomposition with overestimation of rank using a dictionary approach*

In this phase, we compute the CPD of a data sub-tensor $\mathcal{T}_0$ with an overestimated rank. The main idea is to compute the factor matrices $\mathbf{A}$ and $\mathbf{B}$ as the product of sparse dictionaries $\mathbf{D}^A$ and $\mathbf{D}^B$ by atoms $\mathbf{V}^A$ and $\mathbf{V}^B$ respectively:

$$\mathbf{A} = \mathbf{D}^A\mathbf{V}^A \text{ and } \mathbf{B} = \mathbf{D}^B\mathbf{V}^B, \tag{2}$$

where $\mathbf{D}^A$ and $\mathbf{D}^B$ are of size (I, R) and (J, R) respectively and $\mathbf{V}^A$ and $\mathbf{V}^B$ are (R, R). We want that $\mathbf{D}^A$ and $\mathbf{D}^B$ contain $\widetilde{R}_0$ true factors among their columns along with $R-\widetilde{R}_0$ factors without physical meaning. We thus expect that $\mathbf{V}^A$ and $\mathbf{V}^B$ have $R-\widetilde{R}_0$ null columns and that the other columns form a generalized permutation matrix. For this purpose, we solve the following minimization problem for $\mathcal{T} = \mathcal{T}_0$:

$$\begin{aligned} &\min \left\{ E_1(\mathbf{D}^A, \mathbf{V}^A, \mathbf{D}^B, \mathbf{V}^B, \mathbf{C}) \right\} \; s.t. \; \mathbf{D}^A, \mathbf{V}^A, \mathbf{D}^B, \mathbf{V}^B, \mathbf{C} \geq 0 \\ &\text{where } E_1 = \frac{1}{2}\|\mathcal{T} - \widehat{\mathcal{T}}(\mathbf{D}^A\mathbf{V}^A, \mathbf{D}^B\mathbf{V}^B, \mathbf{C})\|_F^2 \\ &+ \alpha\|\mathbf{V}^A\|_1 + \alpha\|\mathbf{V}^B\|_1 \end{aligned} \tag{3}$$

with $\alpha > 0$ a penalty coefficient term. Thereby the number of non-null columns of $\mathbf{A}$ or $\mathbf{B}$ gives us a *posterior* estimation of $\widetilde{R}_0$.

B. Online *phase*

In this phase, we want to compute the CPD of sub-tensor $\mathcal{T}_n$ using the factors ($\mathbf{A}_{n-1}$, $\mathbf{B}_{n-1}$) estimated from $\mathcal{T}_{n-1}$.

a) Approach 1. Online CPD using a dictionary learning approach: For this first approach, we consider that the factors estimated at time t_{n-1} are correct and we only look for the apparition of news factors or the disappearance of old ones. For this purpose, we first set $\mathbf{D}^A = \mathbf{A}_{n-1}$ and $\mathbf{D}^B = \mathbf{B}_{n-1}$. Then we solve Eq. (3) for $\widehat{\mathcal{T}} = \widehat{\mathcal{T}}_n$ but this time we keep the non-null columns of $\mathbf{D}^A$ and $\mathbf{D}^B$ unchanged throughout the optimization process. Indeed, these columns correspond to the factors estimated previously. If one of these factors has disappeared at time t_n we should then obtain a zero in $\mathbf{V}^A$ and $\mathbf{V}^B$. We call this algorithm Online Sparse and Nonnegative CPD 1 (OSNCPD 1). The different steps are summarized in Algorithm 1.

b) Approach 2. Online CPD using a dictionary transfer learning approach: In this approach we allow some modification of the factors estimated at time t_{n-1} thanks to linear combinations as explained below. We first replace all the null columns of $\mathbf{A}_{n-1}$ and $\mathbf{B}_{n-1}$ by columns of random numbers

[1]Both algorithms can be easily extended to higher order tensors.

Algorithm 1 Online Sparse and Nonnegative CPD 1

- **STEP 1: Initialization phase**

Input : $\mathcal{T}_0$, R *overestimated*
Solve Eq. (3) *with* $\mathcal{T} = \mathcal{T}_0$
Compute matrices $\mathbf{A}_0$ *and* $\mathbf{B}_0$ *using Eq.* (2)
Output : $\mathbf{A}_0$, $\mathbf{B}_0$ *and* $\mathbf{C}_0$

- **STEP 2: Online phase**

Input : $\mathcal{T}_n$ *and* $\mathbf{A}_{n-1}, \mathbf{B}_{n-1}$
Initialize $\mathbf{D}^A = \mathbf{A}_{n-1}$, $\mathbf{D}^B = \mathbf{B}_{n-1}$
Fill the null columns of $\mathbf{D}^A$ *and* $\mathbf{D}^B$ *by random numbers, exclude the other columns from the optimization problem Eq.* (3) *and solve it for* $\mathcal{T} = \mathcal{T}_n$
Update current matrices factors : $\mathbf{A}_n = \mathbf{D}^A\mathbf{V}^A$ *and* $\mathbf{B}_n = \mathbf{D}^B\mathbf{V}^B$
Output : $\mathbf{A}_n, \mathbf{B}_n, \mathbf{C}_n$ *and* $\widetilde{R}_n$

- **STEP 3: Return to Step 2 with n=n+1**

drawn from the standard normal distribution. We then look for $\mathbf{A}_n$ and $\mathbf{B}_n$ as:

$$\mathbf{A}_n = \mathbf{U}^A\mathbf{A}_{n-1}\mathbf{V}^A \text{ and } \mathbf{B}_n = \mathbf{U}^B\mathbf{B}_{n-1}\mathbf{V}^B \tag{4}$$

where $\mathbf{U}^A$ and $\mathbf{U}^B$ are two square matrices of sizes I and J respectively. With respect to approach 1, $\mathbf{U}^A\mathbf{A}_{n-1}$ and $\mathbf{U}^B\mathbf{B}_{n-1}$ play the role of flexible dictionaries, while $\mathbf{V}^A$ and $\mathbf{V}^B$ matrices are still sparse atoms. Therefore the optimization problem becomes:

$$\begin{aligned}&\min\left\{E_2(\mathbf{U}^A,\mathbf{V}^A,\mathbf{U}^B,\mathbf{V}^B,\mathbf{C})\right\} \; s.t\; \mathbf{U}^A,\mathbf{U}^B,\mathbf{C},\mathbf{V}^B,\mathbf{V}^B \geq 0\\ &\text{where } E_2 = \frac{1}{2}\|\mathcal{T} - \widehat{\mathcal{T}}(\mathbf{U}^A\mathbf{A}_{n-1}\mathbf{V}^B, \mathbf{U}^B\mathbf{B}_{n-1}\mathbf{V}^B, \mathbf{C})\|_F^2\\ &+\alpha\|\mathbf{V}^A\|_1 + \alpha\|\mathbf{V}^B\|_1\end{aligned} \tag{5}$$

We call this algorithm Online Sparse and Nonnegative CPD 2 (OSNCPD 2). The different steps are summarized in Algorithm 2.

Algorithm 2 Online Sparse and Nonnegative CPD 2

- **STEP 1: Initialisation phase**

Identical to the initialization of the algorithm 1

- **STEP 2: Online phase**

Input : $\mathcal{T}_n$, *and* $\mathbf{A}_{n-1}, \mathbf{B}_{n-1}$
Fill null columns of $\mathbf{A}_{n-1}, \mathbf{B}_{n-1}$ *by random numbers and Solve Eq.* (5) *for* $\mathcal{T} = \mathcal{T}_n$
Update current matrix factors : $\mathbf{A}_n$ *and* $\mathbf{B}_n$ *using Eq.* (4)
Output : $\mathbf{A}_n, \mathbf{B}_n, \mathbf{C}_n$ *and* $\widetilde{R}_n$

- **STEP 3: Return to Step 2 with n=n+1**

C. Optimization scheme

We use a Stochastic Gradient Descent (SGD) algorithm called $Nadam$ [20] in order to solve Eq. (3) or Eq. (5). SGD algorithms have already been used to compute the CPD in [21]. $Nadam$ is a variant of $Adam$, a popular SGD algorithm, less sensitive to the step size [20]. We thus need to compute the gradients of E_1 or E_2. For Eq. (3) we obtain:

$$\begin{aligned}\frac{\partial E_1}{\partial \mathbf{V}^A} &= -(\mathbf{D}^A)^\top(\mathbf{T}^{I,KJ}_{(1)} - \mathbf{D}^A\mathbf{V}^A\mathbf{L}^\top_{(1)})\mathbf{L}_{(1)} + \alpha\mathbf{1}_R\\ \frac{\partial E_1}{\partial \mathbf{V}^B} &= -(\mathbf{D}^B)^\top(\mathbf{T}^{J,KI}_{(2)} - \mathbf{D}^B\mathbf{V}^B\mathbf{L}^\top_{(2)})\mathbf{L}_{(2)} + \alpha\mathbf{1}_R\\ \frac{\partial E_1}{\partial \mathbf{D}^A} &= -(\mathbf{T}^{I,KJ}_{(1)} - \mathbf{D}^A\mathbf{V}^A\mathbf{L}^\top_{(1)})\mathbf{L}_{(1)}(\mathbf{V}^A)^\top\\ \frac{\partial E_1}{\partial \mathbf{D}^B} &= -(\mathbf{T}^{J,KI}_{(2)} - \mathbf{D}^B\mathbf{V}^B\mathbf{L}^\top_{(2)})\mathbf{L}_{(2)}(\mathbf{V}^B)^\top\\ \frac{\partial E_1}{\partial \mathbf{C}} &= -(\mathbf{T}^{K,JI}_{(3)} - \mathbf{C}\mathbf{L}^\top_{(3)})\mathbf{L}_{(3)}\end{aligned} \tag{6}$$

Matrices $\mathbf{T}^{I,KJ}_{(1)}, \mathbf{T}^{J,KI}_{(2)}, \mathbf{T}^{K,JI}_{(3)}$ are obtained by unfolding the tensor $\boldsymbol{\mathcal{T}}$ with respect to the first, second and third modes respectively. Let $\mathbf{L}_{(1)} = \mathbf{C}\odot(\mathbf{D}^B\mathbf{V}^B)$, $\mathbf{L}_{(2)} = \mathbf{C}\odot(\mathbf{D}^A\mathbf{V}^A)$ and $\mathbf{L}_{(3)} = (\mathbf{D}^B\mathbf{V}^B)\odot(\mathbf{D}^A\mathbf{V}^A)$, where operator $\odot$ is the Khatri-Rao product. $\mathbf{1}_R$ is a (R, R) matrix of 1.

For Eq. (5) we get:

$$\begin{aligned}\frac{\partial E_2}{\partial \mathbf{V}^A} &= -(\mathbf{U}^A\mathbf{A}_{n-1})^\top(\mathbf{T}^{I,KJ}_{(1)} - \mathbf{U}^A\mathbf{A}_{n-1}\mathbf{V}^A\mathbf{Z}^\top_{(1)})\mathbf{Z}_{(1)} + \alpha\mathbf{1}_R\\ \frac{\partial E_2}{\partial \mathbf{V}^B} &= -(\mathbf{U}^B\mathbf{B}_{n-1})^\top(\mathbf{T}^{J,KI}_{(2)} - \mathbf{U}^B\mathbf{B}_{n-1}\mathbf{V}^B\mathbf{Z}^\top_{(2)})\mathbf{Z}_{(2)} + \alpha\mathbf{1}_R\\ \frac{\partial E_2}{\partial \mathbf{U}^A} &= -(\mathbf{T}^{I,KJ}_{(1)} - \mathbf{U}^A\mathbf{A}_{n-1}\mathbf{V}^B\mathbf{Z}^\top_{(1)})\mathbf{Z}_{(1)}(\mathbf{V}^A)^\top(\mathbf{A}_{n-1})^\top\\ \frac{\partial E_2}{\partial \mathbf{U}^B} &= -(\mathbf{T}^{J,KI}_{(2)} - \mathbf{U}^B\mathbf{B}_{n-1}\mathbf{V}^B\mathbf{Z}^\top_{(2)})\mathbf{Z}_{(2)}(\mathbf{V}^B)^\top(\mathbf{B}_{n-1})^\top\\ \frac{\partial E_2}{\partial \mathbf{C}} &= -(\mathbf{T}^{K,JI}_{(3)} - \mathbf{C}\mathbf{Z}^\top_{(3)})\mathbf{Z}_{(3)}\end{aligned} \tag{7}$$

with $\mathbf{Z}_{(1)} = \mathbf{C}\odot(\mathbf{U}^B\mathbf{B}_{n-1}\mathbf{V}^B)$; $\mathbf{Z}_{(2)} = \mathbf{C}\odot(\mathbf{U}^A\mathbf{A}_{n-1}\mathbf{V}^A)$ and $\mathbf{Z}_{(3)} = (\mathbf{U}^B\mathbf{B}_{n-1}\mathbf{V}^B)\odot(\mathbf{U}^A\mathbf{A}_{n-1}\mathbf{V}^A)$.

In order to ensure the nonnegativity of matrix entries, all the element of the matrices $\mathbf{D}^A, \mathbf{D}^B, \mathbf{C}, \mathbf{U}^A, \mathbf{U}^B, \mathbf{V}^A$ and $\mathbf{V}^B$ are projected on $\mathbb{R}_+$ at each iteration. For both phases, $\mathbf{V}^A$ and $\mathbf{V}^B$ are initialized as identity matrices and $\mathbf{C}$ is initialized with nonnegative random values drawn from the standard normal distribution. For the *initialization* phase, $\mathbf{D}^A$ and $\mathbf{D}^B$ are initialized with nonnegative random values drawn from the standard normal distribution. For the *online* phase of Algorithm 2, $\mathbf{U}^A$ and $\mathbf{U}^B$ are initialized with the identity matrix.

III. ONLINE DECOMPOSITION OF REAL FLUORESCENCE TENSORS

A. The NN-CPD model of real-time fluorescence measurements

In environmental sciences, the CPD of three-way fluorescence tensors is notably used in order to characterize dissolved organic matters (DOM) in natural water samples [22]. Fluorescence data tensors are built by concatenation of 2D signals called Emission and Excitation Matrices (EEMs) of fluorescence measured from a set of liquid samples. Each entry of an EEM corresponds to the fluorescence intensity of one liquid sample at a given couple of excitation and emission wavelengths. Each sample is a mixture of an unknown number of fluorescent components (fluorophores). The CPD of the fluorescence tensor allows to recover the individual emission and excitation spectra of the fluorophores present in the different samples along with their respective contribution, in order to characterize and track online the

chemical components present in the sample set. Online CPD of fluorescence tensors means that several fluorescence sub-tensors are measured successively at different time intervals, assuming that each new tensor is a slight modification of the previous one (Fig. 1). Therefore we can use the result of the previous CPD to compute the CPD of the current tensor. The number of fluorophores present at time t_n defines the physical rank of the corresponding fluorescence sub-tensor. It can vary with time and these variations reflect changes in the environment. Online CPD of fluorescence tensors could be used for environmental monitoring, detection of pollution, etc.

B. Experimental setup

In order to evaluate our approaches under similar conditions we performed in our laboratory a series of controlled injections of four well-known fluorophores at different time intervals under quasi-real conditions. 50 EEMs of size $(49, 44)$ were recorded online with a spectrofluorimeter *Hitachi F7000*. The corresponding fluorescence tensor was partitioned into 4 successive sub-tensors ($\mathcal{T}_0 ... \mathcal{T}_3$). We considered two kinds of partitions. In the first partition (50% overlapping), each sub-tensor contains 20 EEMs and two consecutive sub-tensors have 10 EEMs in common. In the second partition (no overlapping), sub-tensors have no EEMs in common: $\mathcal{T}_0$ contains 20 EEMs and the other sub-tensors contain only 10 EEMs. The second partition should make the decomposition more difficult. However, it is more interesting from a practical point of view since it requires less computations as the online phase deals with smaller tensors. During the acquisition, we observed the presence of an additional unexpected fluorescence signal in all the EEMs due to the experimental device. A preliminary study showed that this signal can be represented by a single additional factor in the CPD model. Therefore, the rank of the sub-tensors can be 3, 4 or 5. Both partitions were processed with OSNCPD 1 and OSNCPD 2. We chose $R = 10$ in order to overestimate the rank.

C. Results and discussion

We first compare the values of the physical ranks estimated by OSNCPD1 and OSNCPD2 with those estimated by AutoTen [9] and the NN-CPD algorithm proposed in [11] and the actual values. Results are reported in Tab. I and II for the 50% overlapping case and no overlapping case, respectively. We also indicate the value of the penalty coefficient term (α) used in our two algorithms and in NN-CPD. AutoTen fails for two sub-tensors in the 50% overlapping case and consistently underestimates the rank in the no overlapping case. This is explained by the weaker contribution of several fluorophores in the last EEMs and the correlation of the spectra. Conversely, NN-CPD overestimated the rank in the last sub-tensors in both cases. In contrast, OSNCPD 1 and OSNCPD 2 find the correct rank of every sub-tensors in both cases.

Removing null columns of $\widehat{\mathbf{A}}$ and $\widehat{\mathbf{B}}$ and the corresponding columns of $\widehat{\mathbf{C}}$, we can compare the factor matrices estimated by OSNCPD 1 and OSNCPD 2 with the true factors by means of normalized root mean squared errors:

$$E_A = \frac{\|\mathbf{A} - \widehat{\mathbf{A}}\|_F}{\|\mathbf{A}\|_F},\ E_B = \frac{\|\mathbf{B} - \widehat{\mathbf{B}}\|_F}{\|\mathbf{B}\|_F} \text{ and } E_C = \frac{\|\mathbf{C} - \widehat{\mathbf{C}}\|_F}{\|\mathbf{C}\|_F} \tag{8}$$

Results are reported in Tab. III and IV for the 50% overlapping case and no overlapping case respectively.

Table I
RANK ESTIMATION IN THE OVERLAPPING CASE.

Sub-tensor	**0**	**1**	**2**	**3**	α
True rank	**3**	**4**	**5**	**5**	-
AutoTen	**3**	**4**	4	4	-
NN-CPD	**3**	**4**	7	10	0.5 - 1
OSNCPD1 & **OSNCPD2**	**3**	**4**	**5**	**5**	1

Table II
RANK ESTIMATION IN THE NO OVERLAPPING CASE.

Sub-tensor	**0**	**1**	**2**	**3**	α
True rank	**3**	**4**	**5**	**5**	-
AutoTen	3	2	2	2	-
NN-CPD	**3**	**4**	**5**	10	0.1 - 0.5
OSNCPD1 & **OSNCPD2**	**3**	**4**	**5**	**5**	0.5

Table III
MEAN ESTIMATIONS ERRORS IN THE OVERLAPPING CASE.

	Algorithm 1 (OSNCPD1)				**Algorithm 2 (OSNCPD2)**			
Sub-tensor	**0**	**1**	**2**	**3**	**0**	**1**	**2**	**3**
Mean E_A	0.17	0.19	0.19	0.2	0.17	0.18	0.2	0.34
Mean E_B	0.14	0.15	0.15	0.16	0.14	0.16	0.19	0.24
Mean E_C	0.25	0.25	0.16	0.20	0.25	0.26	0.18	0.29

Table IV
MEAN ESTIMATIONS ERRORS IN THE NO OVERLAPPING CASE.

	Algorithm 1 (OSNCPD1)				**Algorithm 2 (OSNCPD2)**			
Sub-tensor	**0**	**1**	**2**	**3**	**0**	**1**	**2**	**3**
Mean E_A	0.17	0.28	0.35	0.35	0.17	0.18	0.21	0.33
Mean E_B	0.14	0.15	0.31	0.32	0.14	0.16	0.19	0.24
Mean E_C	0.25	0.32	0.31	0.32	0.25	0.31	0.32	0.29

In order to give some physical meaning to these error terms that may seem high, we have plotted on Fig. 2 and 3 the normalized factors obtained from our approach in the initialization phase (sub-tensor $\mathcal{T}_0$) along with the true factors. We observe a good agreement between the true and estimated factors. However there is an offset in the estimated contributions. We explain this deviation by the fact that the "true" contributions are not exactly known since they were established theoretically from the characteristics of our injection and evacuation devices. In addition this theoretical model does not take into account the additional fluorophore. This offset explains the higher values of E_C with respect to E_A and E_B. Eventually, the estimated contribution of the additional fluorophore is constant as it was expected for a background noise.

In the overlapping case both algorithms give similar results for sub-tensors $\mathcal{T}_1$ and $\mathcal{T}_2$. In addition these are close to those obtained from the initialization sub-tensor $\mathcal{T}_0$ meaning that the online phase has been performed correctly. Regarding $\mathcal{T}_3$, OSNCPD1 still works well whereas the performances of OSNCPD2 $\mathcal{T}_3$ have decreased. This is explained by the very weak contributions of some factors in $\mathcal{T}_3$. Here, OSNCPD1 has taken advantages of its strongest link with the previous CPD.

In the no overlapping case OSNCPD2 consistently outperforms OSNCPD1 especially for sub-tensors $\mathcal{T}_2$ and $\mathcal{T}_3$. Here

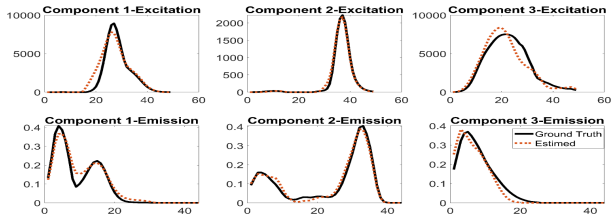

Figure 2. Emission and excitation and spectra of the fluorophores in the initialization phase ($\mathcal{T}_0$). Top: Emission spectra, bottom : Excitation spectra. Red dots: estimated spectra, black lines: true spectra. All spectra are normalized.

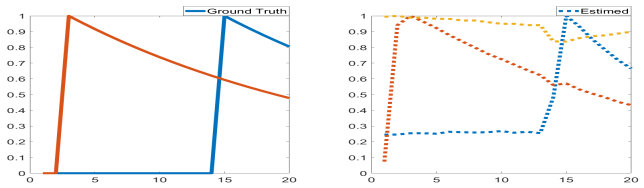

Figure 3. Fluorophore contributions in the initialization phase ($\mathcal{T}_0$). Top: "true" contributions (the contribution of the additional fluorophore is unknown). Bottom: estimated contributions. All contributions are normalized.

it appears that the value of E_A obtained with OSNCPD1 on $\mathcal{T}_1$ is already quite high and then the error propagates to the successive sub-tensors. This is probably due to the small size of the sub-tensors. In this context the greater flexibility of OSNCPD2 alleviates this problem. Regarding the estimation of **A** and **B**, it is worth mentioning that OSNCPD2 results do not vary from the overlapping case to the no overlapping case.

IV. CONCLUSION & PERSPECTIVES

In this study, we have introduced two algorithms (OSNCPD1 and OSNCPD2) for the online nonnegative canonical polyadic decomposition of sub-tensors and a real world application example in fluorescence signal processing. Our algorithms are based on dictionary learning and incorporate sparsity and nonnegativity constraints in order to deal with unknown rank variations. We have shown that both algorithms can be successfully applied on real experimental data. In particular, the ranks of the sub-tensors are estimated correctly even in difficult conditions. It seems that OSNCPD1 is more robust than OSNCPD2 when an overlapping between consecutive sub-tensors is possible. In contrast OSNCPD2 offers a more versatile solution, able to deal with the no overlapping case. Further investigations are planned to test these conclusions in diverse situations, evaluate the influence of the penalty term and use the two first modes conjointly to estimate the rank. In addition, experiments are planned to compare our approaches and the rest of the methods proposed in the state of the art with a fixed rank.

ACKNOWLEDGMENT

We gratefully acknowledge Region Sud PACA (France) and BIOCEANOR for their support.

REFERENCES

[1] F. L. Hitchcock, "Multiple invariants and generalized rank of a p-way matrix or tensor," *J. Math. and Phys.*, vol. 7, no. 1, pp. 39–79, 1927.

[2] R. A. Harshman, "Foundation of PARAFAC procedure: Models and conditions for an 'explanatory' multi-mode factor analysis," *UCLA working papers in Phonetics*, no. 16, pp. 1–84, 1970.

[3] A. S. Field and D. Graupe, "Topographic component (parallel factor) analysis of multichannel evoked potentials: practical issues in trilinear spatiotemporal decomposition," *Brain topography*, vol. 3, no. 4, pp. 407–423, 1991.

[4] R. Bro, "PARAFAC. tutorial and applications," *Chemometrics and intelligent laboratory systems*, vol. 38, no. 2, pp. 149–171, 1997.

[5] I. Kopriva and A. Cichocki, "Blind multispectral image decomposition by 3D nonnegative tensor factorization," *Optics letters*, vol. 34, no. 14, pp. 2210–2212, 2009.

[6] G. Tomasi and R. Bro, "A comparison of algorithms for fitting the PARAFAC model," *Computational Statistics & Data Analysis*, vol. 50, no. 7, pp. 1700–1734, 2006.

[7] K. Huang, N. D. Sidiropoulos, and A. P. Liavas, "A flexible and efficient algorithmic framework for constrained matrix and tensor factorization," *IEEE Transactions on Signal Processing*, vol. 64, no. 19, pp. 5052–5065, 2016.

[8] R. Bro and H. A. Kiers, "A new efficient method for determining the number of components in PARAFAC models," *Journal of Chemometrics*, vol. 17, no. 5, pp. 274–286, 2003.

[9] E. E. Papalexakis, "Automatic unsupervised tensor mining with quality assessment," in *Proceedings of the 2016 SIAM International Conference on Data Mining*, pp. 711–719, SIAM, 2016.

[10] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM Journal on imaging sciences*, vol. 6, no. 3, pp. 1758–1789, 2013.

[11] J.-P. Royer, N. Thirion-Moreau, P. Comon, R. Redon, and S. Mounier, "A regularized nonnegative canonical polyadic decomposition algorithm with preprocessing for 3d fluorescence spectroscopy," *Journal of Chemometrics*, vol. 29, no. 4, pp. 253–265, 2015.

[12] X. Vu, C. Chaux, N. Thirion-Moreau, S. Maire, and E. M. Carstea, "A new penalized nonnegative third order tensor decomposition using a block coordinate proximal gradient approach: application to 3D fluorescence spectroscopy," *Journal of Chemometrics*, vol. 31, p. e2859, Apr. 2017.

[13] X. Han, L. Albera, A. Kachenoura, L. Senhadji, and H. Shu, "Low rank canonical polyadic decomposition of tensors based on group sparsity," in *2017 25th European Signal Processing Conference (EUSIPCO)*, pp. 668–672, IEEE, 2017.

[14] J. E. Cohen and N. Gillis, "Dictionary-based tensor canonical polyadic decomposition," *IEEE Transactions on Signal Processing*, vol. 66, p. 1876–1889, Apr 2018.

[15] R. André, X. Luciani, L. Albera, and E. Moreau, "A two-step algorithm for joint eigenvalue decomposition-application to canonical polyadic decomposition of fluorescence spectra," *Chemometrics and Intelligent Laboratory Systems*, vol. 206, p. 104065, 2020.

[16] D. Nion and N. D. Sidiropoulos, "Adaptive algorithms to track the PARAFAC decomposition of a third-order tensor," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2299–2310, 2009.

[17] S. Zhou, X. V. Nguyen, J. Bailey, Y. Jia, and I. Davidson, "Accelerating online CP decompositions for higher order tensors," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1375–1384, ACM, 2016.

[18] R. Pasricha, E. Gujral, and E. E. Papalexakis, "Identifying and alleviating concept drift in streaming tensor decomposition," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 327–343, Springer, 2018.

[19] A. Traoré, M. Berar, and A. Rakotomamonjy, "Online multimodal dictionary learning," *Neurocomputing*, vol. 368, pp. 163–179, 2019.

[20] T. Dozat, "Incorporating NESTEROV momentum into ADAM," *Proceedings of 4th International Conference on Learning Representations, Workshop Track*, 2016.

[21] N. Vervliet and L. De Lathauwer, "A randomized block sampling approach to canonical polyadic decomposition of large-scale tensors," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 2, pp. 284–295, 2015.

[22] K. R. Murphy, C. A. Stedmon, D. Graeber, and R. Bro, "Fluorescence spectroscopy and multi-way techniques.PARAFAC," *Analytical Methods*, vol. 5, no. 23, pp. 6557–6566, 2013.